Networked Embedded Systems WS 2016/17

Lecture 1: Introduction

Marco Zimmerling

TECHNISCHE UNIVERSITÄT DRESDEN

CENTER FOR ADVANCING ELECTRONICS DRESDEN

Credits

The slides presented in this and the following lectures are partially based on material from Giorgio Buttazzo, Björn Brandenburger, Prabal Dutta, Koen Langendoen, Rupak Majumdar, Peter Marwedel, Felix Sutton, and Lothar Thiele.

Today’s Agenda

- Introduction to networked embedded systems
  - What they are
  - Why they are interesting
  - How knowing about them can help you

- Organization
  - Topics and schedule
  - Instructors and course material
  - Lab logistics

Population growth

Urbanization

Climate change

"The world population is projected to increase [from 7.3 billion today] to 9.7 billion in 2050"

United Nations, World Population Prospects: The 2015 Revision

Population growth

Urbanization

Climate change

“By 2050, 66% of the world’s population is projected to be urban”

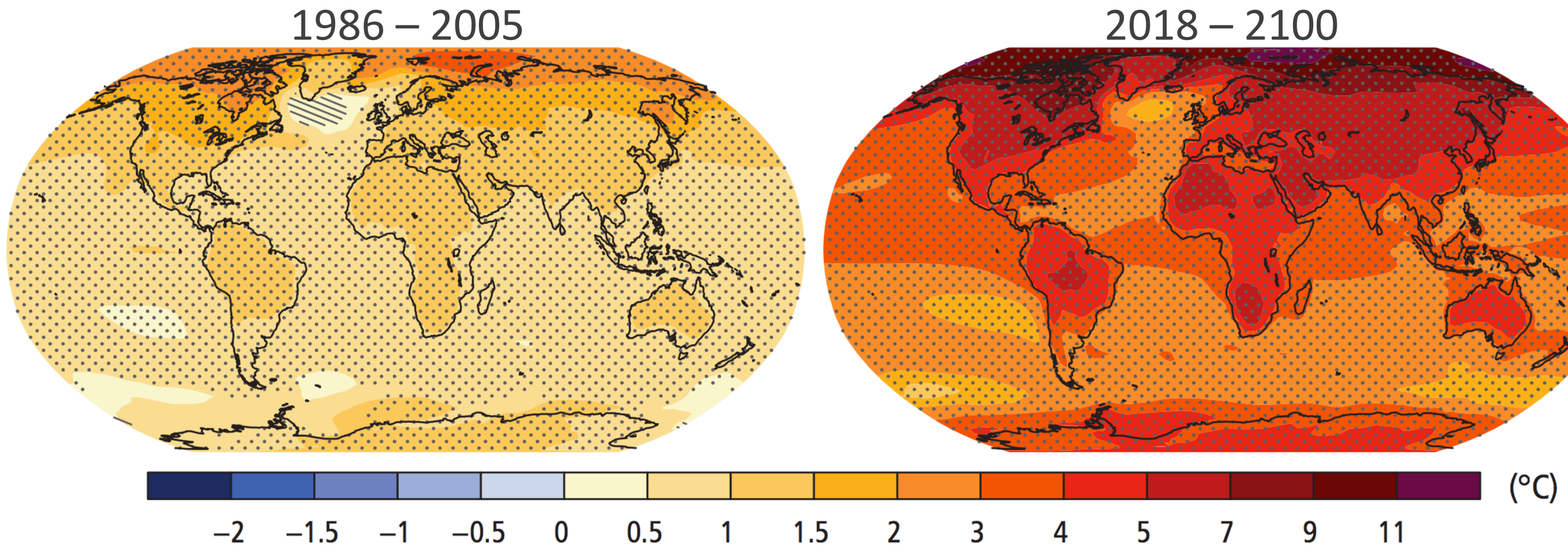
United Nations, World Urbanization Prospects: The 2014 Revision

Population growth

Urbanization

Climate change

Change in average surface temperature



IPCC, Climate Change 2014: Synthesis Report

Population growth

Urbanization

Climate change

These are some of the 21st century's *grand challenges*

and here are potential contributions to solving them …

National Academy of Engineering, National Science Foundation, European Commission

Population growth

Urbanization

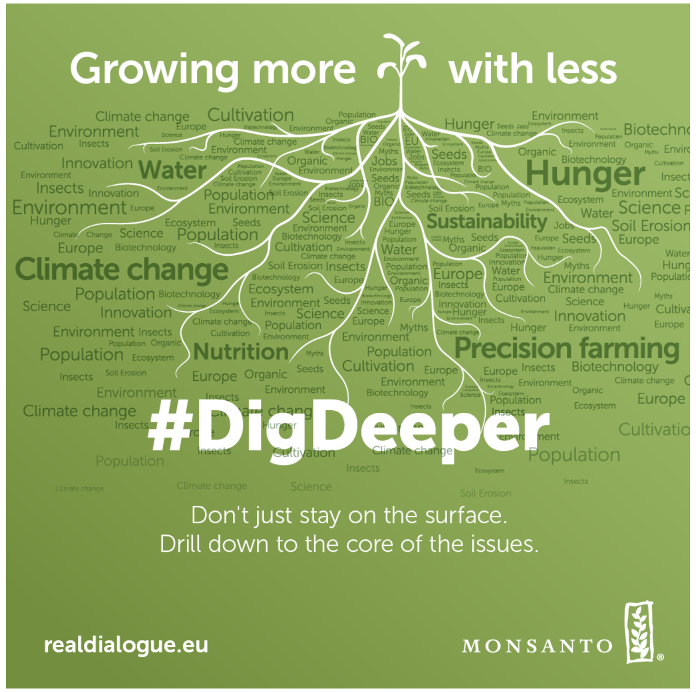
Climate change

Precision agriculture

Smart cities

Disaster mitigation

Precision Agriculture



- Current agricultural practices waste large amounts of water, fertilizer, and pesticides, causing environmental impact and economic losses.
- *Precision agriculture* aims to both reduce waste and increase crop yield using aerial drones, autonomous vehicles, etc.
  - Involves distributed sensing, intelligent real-time decision making, actuation

Smart Cities

THINK!
SMART CITY
DRESDEN
7. November 2014
Ein interdisziplinärer Dialog
im Wissenschaftsjahr 2014
- Die digitale Gesellschaft -

- *Smart cities* aim to optimize processes within large urban areas to, for example, reduce air pollution and improve quality of life.
- Example smart city applications include intelligent transportation, home automation, adaptive lighting, and connected signage.
  - Involves distributed sensing, intelligent real-time decision making, actuation

Disaster Mitigation

- *Disaster mitigation* aims to understand destructive physical processes to predict hazards, mitigating harm to humans and infrastructure.
- An example disaster mitigation application is understanding thawing permafrost in high-alpine regions to possibly predict rockfall events.
  - Involves distributed sensing, intelligent real-time decision making, actuation

What do all these applications have in common?

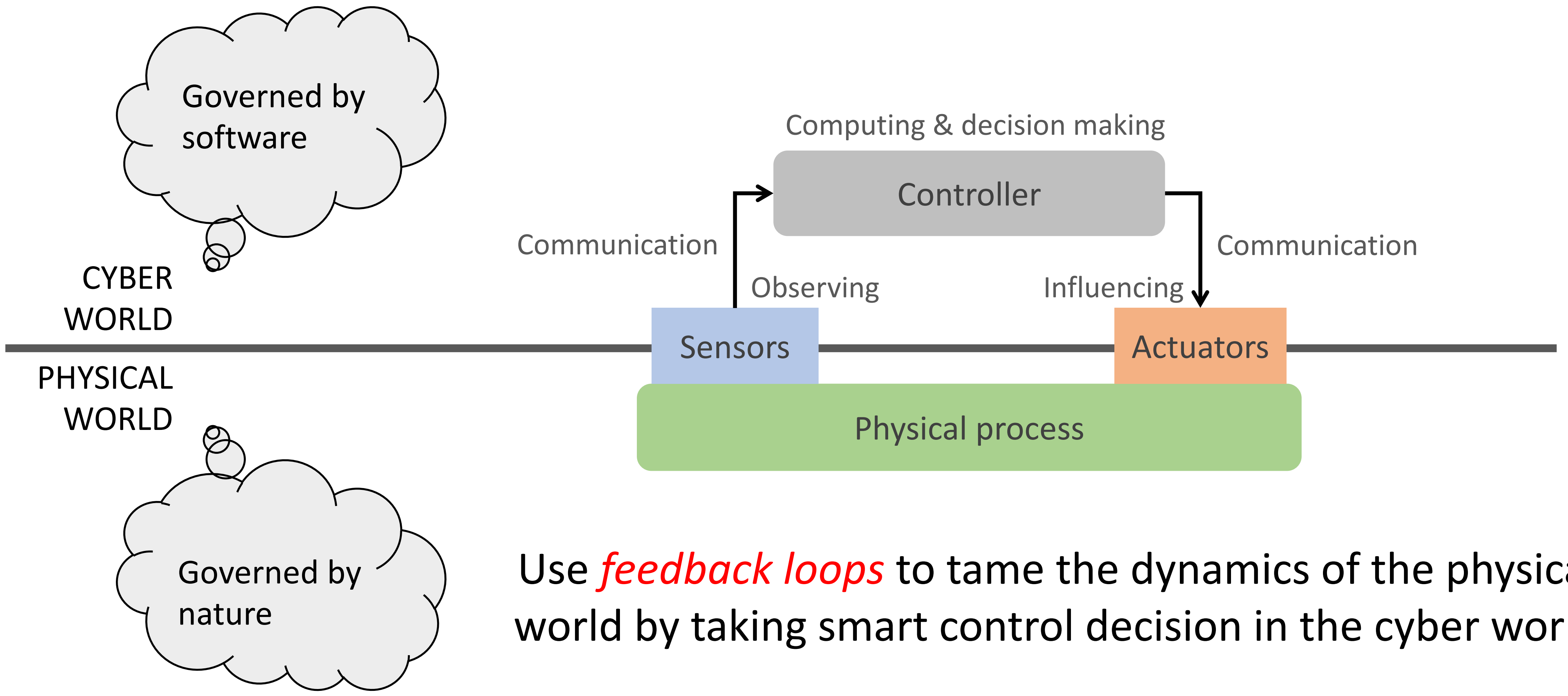
They are *cyber-physical systems (CPS)*

Precision agriculture

Smart cities

Disaster mitigation

Cyber-physical Systems (CPS): Overview



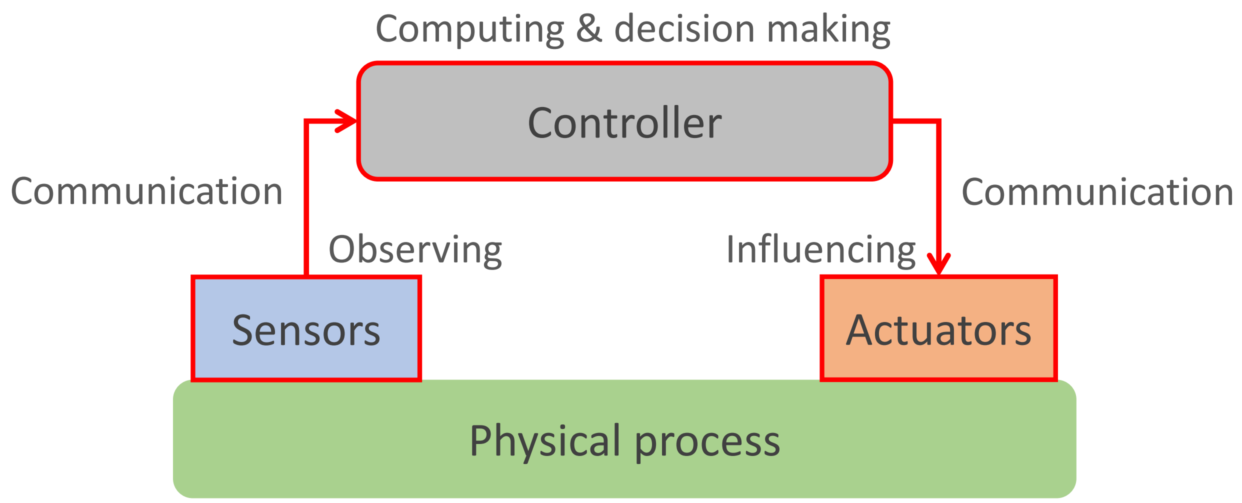
Use *feedback loops* to tame the dynamics of the physical world by taking smart control decision in the cyber world

Cyber-physical Systems (CPS): Definitions

- [Lee07]: “A cyber-physical system (CPS) is an integration of computation with physical processes.”
- [Raj10]: “Cyber-physical systems (CPS) are physical and engineered systems whose operations are monitored, coordinated, controlled and integrated by a computing and communication core.”
- [Der11]: ”Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa. The design of such systems, therefore, requires understanding the joint dynamics of computers, software, networks, and physical processes.”

[Lee07] E. Lee, Computing Foundations and Practice for Cyber-Physical Systems: A Preliminary Report, 2007.
[Raj10] R. Rajkumar et al., *Cyber-Physical Systems: The Next Computing Revolution*, Proc. of DAC, 2010.
[Der11] P. Derler at al., *Modeling Cyber-Physical Systems*, Proc. of the IEEE, 2011.

Networked Embedded Systems



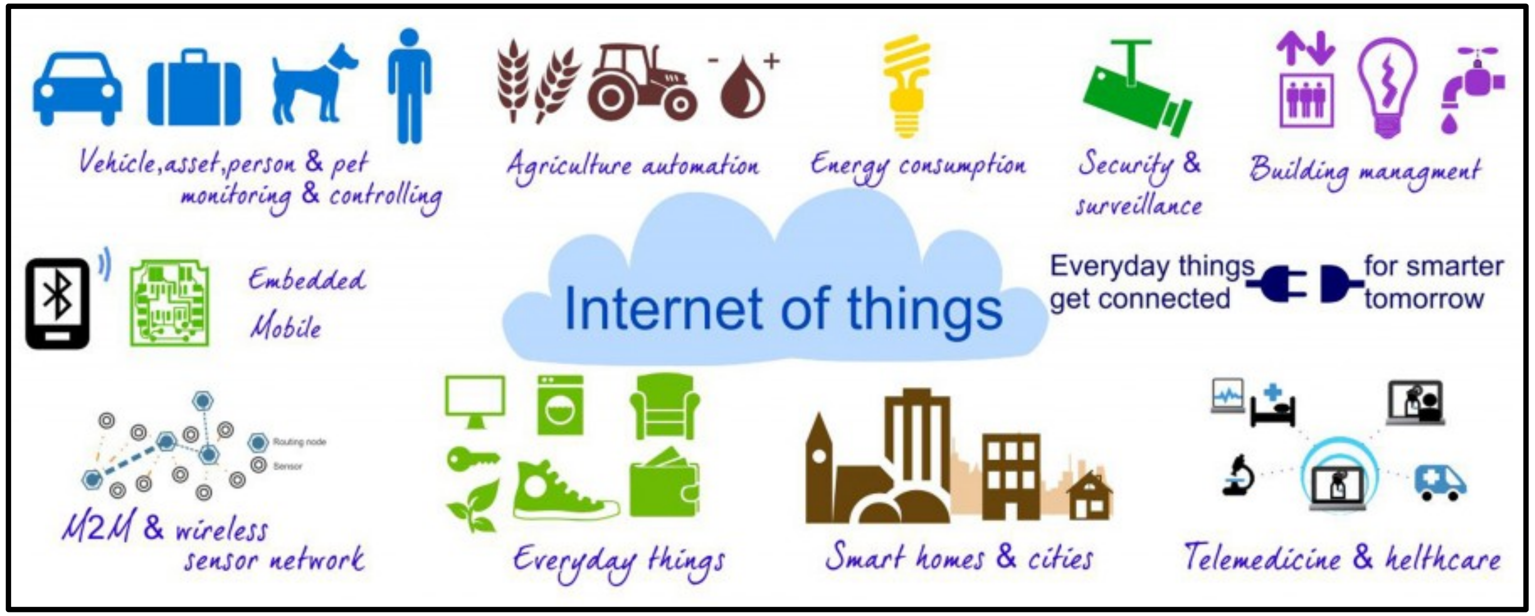
Networked embedded systems represent *parts of the computing and communication infrastructure* in terms of hardware and software *that enable cyber-physical systems*.

“Classical” Networked Embedded Systems

- Anti-lock braking system (ABS)
- Electronic stability control (ESP)
- Airbags
- Automatic gearbox
- Smart keys

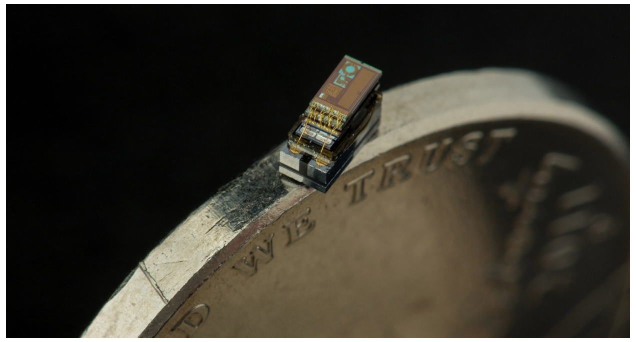
- Flight control system
- Anti-collision control
- Pilot information system
- Flap control system
- Entertainment system

Emerging Networked Embedded Systems



© http://www.satiztpm.it/

Everywhere Wireless


"Smart dust"

- The Internet of Things (IoT) will be omnipresent, consisting of 20-30 billion devices by 2020, the vast majority of which will be wireless.
- Trends: energy harvesting, wireless energy transfer, embedded multi-processors, non-volatile storage, machine learning, data analytics, etc.

Requirements

- CPS often fulfill safety-critical functions and need to face severe resource constraints (*e.g.*, energy on an autonomous aerial drone).
  - [Maj14]: "It is essential to *predict* how a CPS is going to behave under any circumstances […] *before* it is deployed."
  - [Raj10]: "CPS must *operate dependably*, safely, securely, *efficiently* and in real-time."
- Since networked embedded computing and communication enable CPS applications, they are necessarily subject to the exact same *predictability*, *dependability*, and *efficiency* requirements.

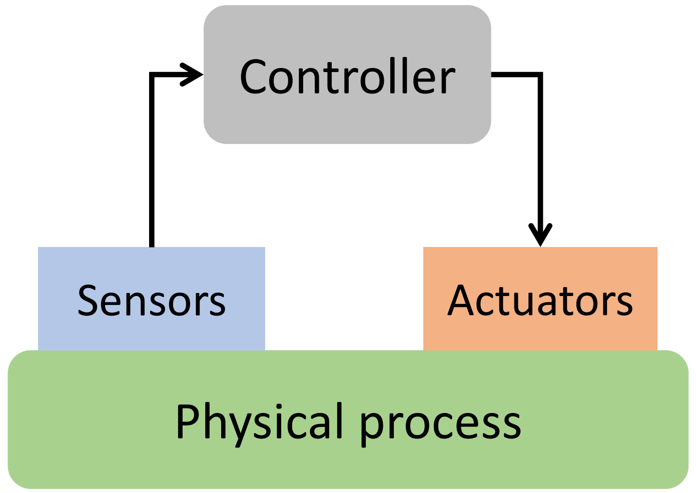
[Maj14] R. Majumdar and B. Brandenburger, *Foundations of Cyber-Physical Systems*, 2014.
[Raj10] R. Rajkumar et al., *Cyber-Physical Systems: The Next Computing Revolution*, Proc. of DAC, 2010.

Predictability

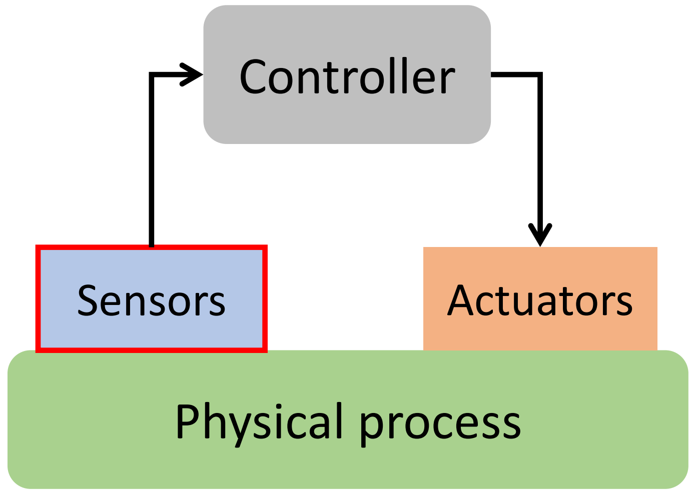
- Comes in different flavors, such as
  - *Timing* (*e.g.*, guarantee that tasks are executed before specified deadlines)
  - *Performance* (*e.g.*, guarantee that the system operates for a certain time)
  - *Functional properties* (*e.g.*, guarantee that packets arrive in a certain order)

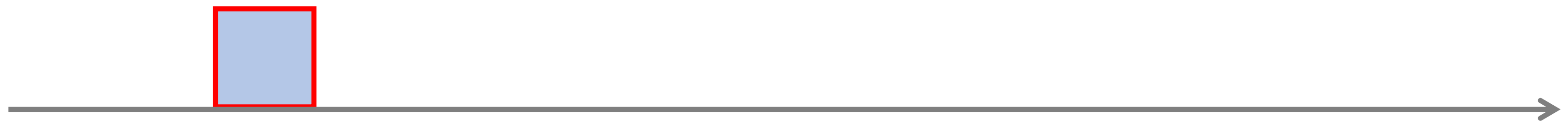
- Timing predictability, that is, the ability to meet *real-time constraints*, is important to keep up with the evolution of the environment over time.
  - Correct behavior of a *real-time system* depends not only on the output of a computation but also on the *time* at which the output is produced.
  - A correct result that arrives too late is wrong nevertheless.

Example: Timing Predictability

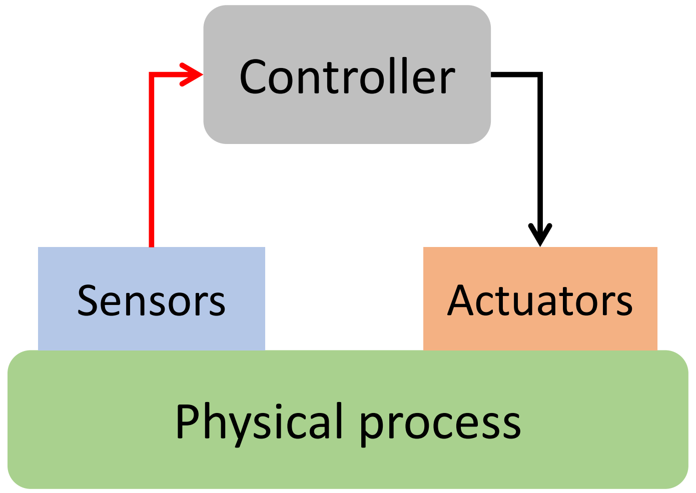


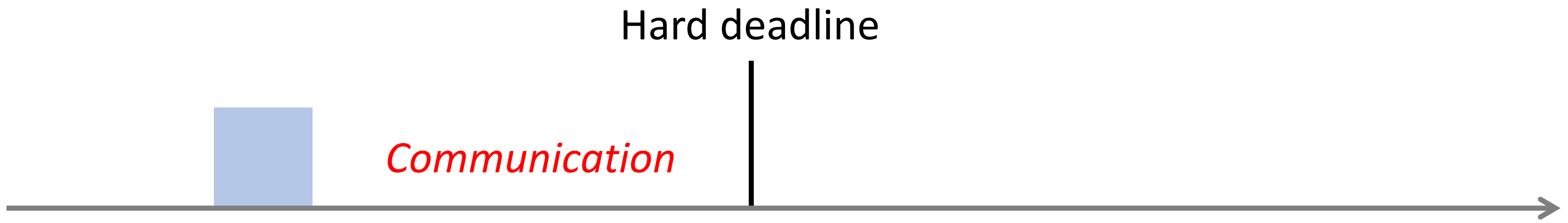
Example: Timing Predictability



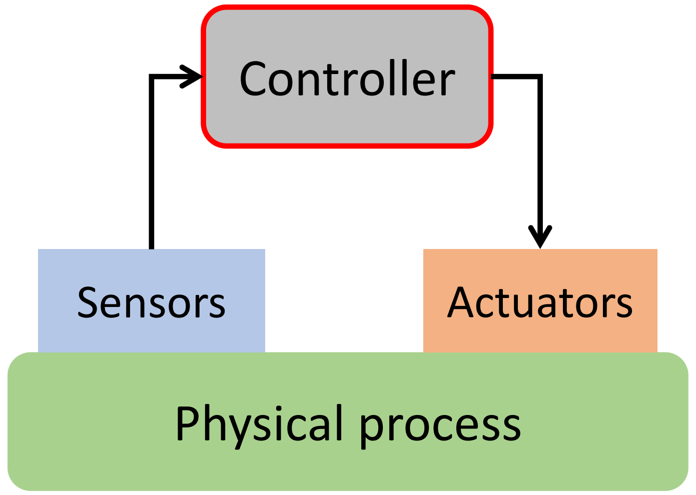


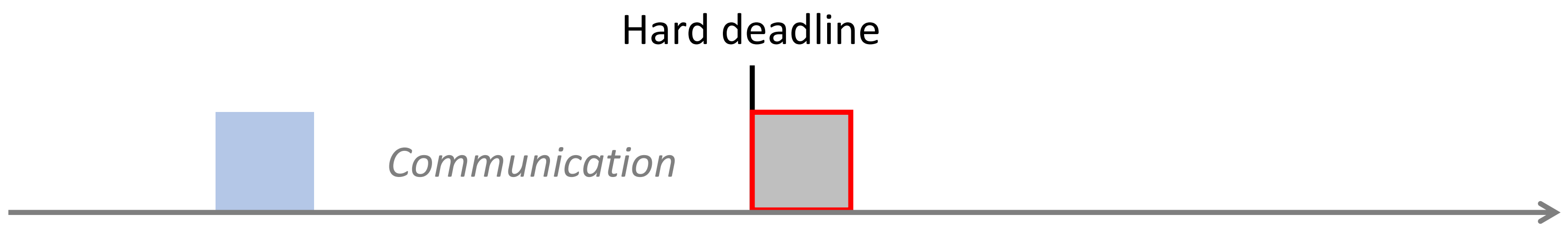
Example: Timing Predictability



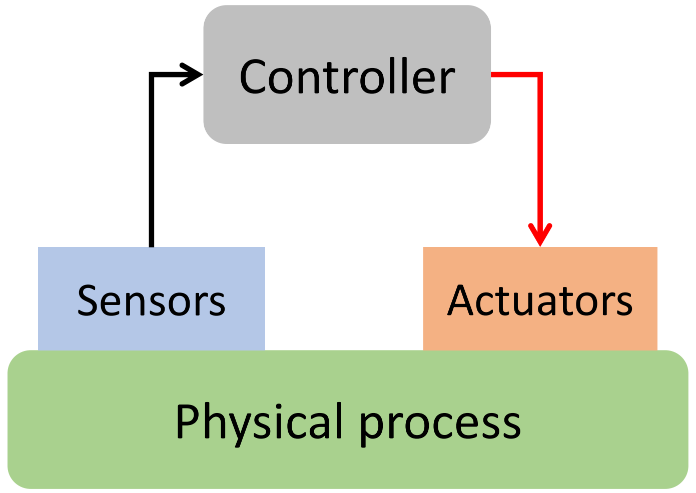


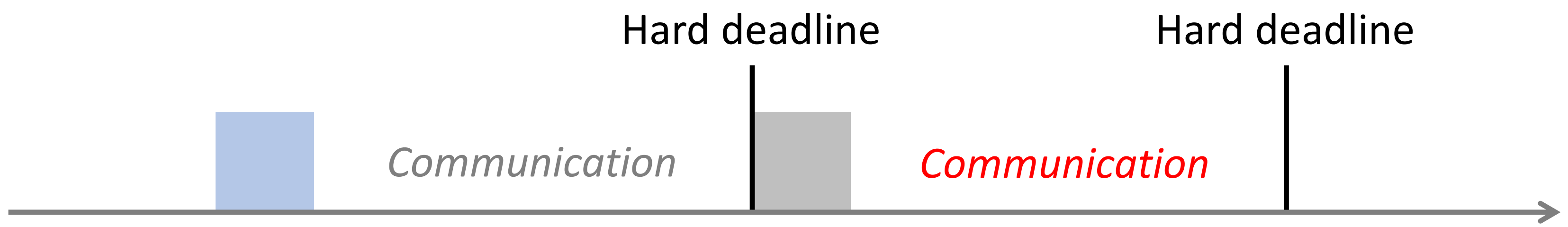
Example: Timing Predictability



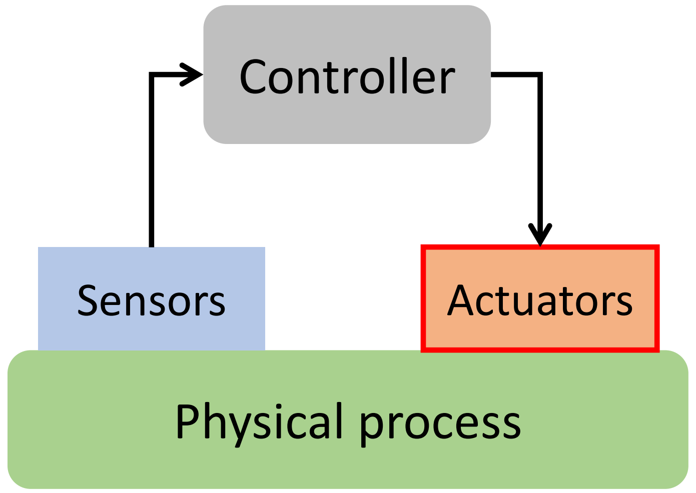


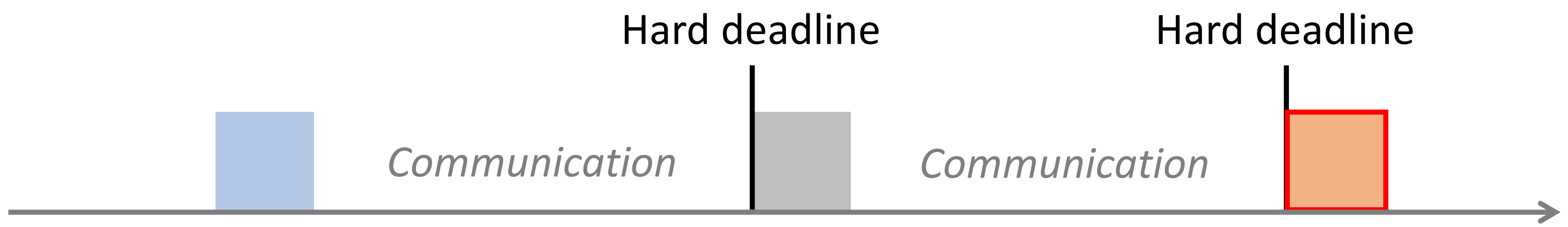
Example: Timing Predictability





Example: Timing Predictability





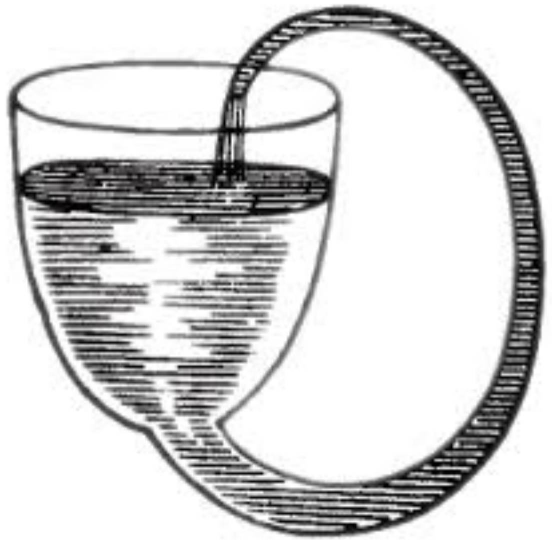
Dependability

- Includes several qualities:
  - Reliability $R(t)$: probability of system working correctly provided that it was working at time $t = 0$.
  - Maintainability $M(d)$: probability of system working correctly d time units after an error occurred.
  - Availability $A(t)$: probability of system working at time t.
  - Safety: system does not harm the users or the environment.
  - Security: confidential and authentic operation and communication.

- Making a networked embedded system dependable must not be an after-thought – it *must be considered from the very beginning*.

Efficiency

- Application requirements and platform constraints call for efficiency
  - Small form factors
  - Small weight
  - Small code size
  - Fast execution
  - Low dynamic memory usage
  - Low wireless bandwidth usage
  - Energy-efficient operation and communication



- Achieved by *tailoring* the system to the specific application needs.

Distinction from General-purpose Systems

- Networked embedded systems are *designed for a single application or a specific, small class of applications* with similar characteristics.
  - General-purpose systems address the needs of a broad class of applications and can be seamlessly reconfigured for a new purpose.
- Networked embedded systems are typically *programmed by highly skilled experts* with detailed knowledge about the system's internals.
  - General-purpose systems typically provide high-level application programming interfaces, allowing end users to program them.
- Networked embedded systems care about *both correct results and that results become available by a certain time* (earlier is not better).
  - General-purpose systems only care correct results; results that become available faster are usually considered better.

Today's Agenda

- Introduction to networked embedded systems
  - What they are
  - Why they are interesting
  - How knowing about them can help you
- Organization
  - Topics and schedule
  - Instructors and course material
  - Lab logistics

Organization: Topics and Schedule

- Focus on three key aspects of networked embedded systems
  - Networked: communication (mostly wireless)
  - Embedded systems: real-time scheduling (of compute tasks)
  - Hands-on experience: programming an embedded wireless platform

- Schedule:
  - Today (Dec 7): Introduction, real-time scheduling (lecture)
  - Friday (Dec 9): Communication (lecture), real-time scheduling (exercise)
  - Wednesday (Dec 14): Lab
  - Friday (Dec 16): Communication (exercise), Q&A, thesis opportunities

Organization: Course Instructors

- Lectures and exercises: Marco Zimmerling
  - Room: BAR I-56
  - Email: marco.zimmerling@tu-dresden.de

- Lab assistants: Fabian Mager / Martina Brachmann
  - Room: BAR I-56A / APB 1039
  - Email: fabian.mager@tu-dresden.de / martina.brachmann@tu-dresden.de

- Please, do not hesitate to ask us during or after the sessions!

Organization: A Note on Course Material

- All course material (slides, task sheets, sample solutions, etc.) will be available on the course website as soon as possible after each session

  https://wwwpub.zih.tu-dresden.de/~mzimmerl/eti.html

- Task sheets will be handed out in print before the exercises and the lab.
  - You will have sufficient time to solve the problem sets yourself.
  - We will be available to assist you and answer your questions.
  - We will present sample solutions in class.
  - We will make these sample solutions available online after each session.

Literature on (Networked) Embedded Systems

- Peter Marwedel, *Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems*, Second Edition, Springer, 2011
- Edward Lee and Sanjit Seshia, *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*, First Edition, 2011 (freely available at http://leeseshia.org/releases/LeeSeshia_DigitalV1_08.pdf)

- Pointers to recommended literature on specific topics are provided at the beginning of each lecture

Organization: Lab Logistics

- Hands-on experience in embedded wireless systems development
  - Hardware: TelosB sensor node (2.4 GHz radio, 8 MHz microcontroller, etc.)
  - Software: Contiki open-source operating system (http://www.contiki-os.org/)
  - You will learn how to read out sensors, how to program an interrupt service routine, and how to send and receive messages over the wireless radio.

- But we need your help!
  - Form groups of two, with one laptop per group.
  - Ensure Internet connectivity (*e.g.*, via eduroam).
  - Preferably Linux, but this is *not* a requirement.